

# TouchKey 2 click

PID: MIKROE-2474



**TouchKey 2 click** has four capacitive pads powered by ATtiny817 which has an integrated touch QTouch® controller. The click is designed to run on a 3.3V power supply. The four LEDs onboard the click indicate when the Key (Pad) is pressed. TouchKey 2 click communicates with the target microcontroller over UART interface (SPI interface is optional).

## High moisture tolerance

You can use TouchKey 2 click in all kinds of conditions, without fearing something will happen to it due to moisture and water droplets falling on it.

The plastic overlay on the TouchKey 2 click protects the board from moisture. Thanks to this feature the electronic components are safe. The ATtiny817 has a driven shield for improved moisture and noise handling performance.

## ATtiny817 features

The Atmel® ATtiny817 is a microcontroller that uses an 8-bit AVR® processor with hardware multiplier, running at up to 20MHz and with up to 8KB Flash, 512 bytes of SRAM and 128 bytes of EEPROM.

ATtiny817 uses the latest technologies from Atmel® with a flexible and low power architecture including Event System and SleepWalking, accurate analog features and advanced peripherals.

Capacitive touch interfaces with proximity sensing and driven shield are supported with the integrated QTouch® peripheral touch controller.

The module supports wake-up on touch from power-save sleep mode.

### Button features

Capacitive buttons can be toggled even when placed under a layer of glass or paper.

### How it works

There are four LEDs for four touch keys. If key A is pressed LED\_A is ON, etc. In addition, there is UART communication between ATtiny817 and main MCU.

**Note:** The header onboard the TouchKey 2 click can be used for device programming. Current firmware sends data packets via UART (on which the demo example in our library is based). **SPI communication** is possible with firmware modifications.

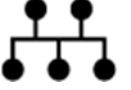
### Key features

- ATtiny817 module
  - Atmel® AVR® 8-bit CPU
  - Running at up to 20MHz
  - 128B EEPROM
  - 256/512B SRAM
  - 32.768kHz external crystal oscillator
- UART interface (SPI optional)
- 3.3V power supply

|                  |   |
|------------------|---|
| Product Type     | Capacitive  |
| Applications     | A replacement for mechanical buttons                                      |
| On-board modules | ATtiny817 from Microchip  |
| Key Features     | Four capacitive touch pads, 3.3V power supply, protective plastic overlay |
| Key Benefits     | High moisture tolerance   |
| Interface        | UART (SPI is optional)  |
| Power Supply     | 3.3V  |
| Compatibility    | mikroBUS  |
| Click board size | L (57.15 x 25.4 mm)   |

## Pinout diagram

This table shows how the pinout on **TouchKey 2 click** corresponds to the pinout on the mikroBUS™ socket (the latter shown in the two middle columns).

| Notes                            | Pin              | <br>mikroBUS™ |      |     |    | Pin          | Notes         |
|----------------------------------|------------------|--|------|-----|----|--------------|---------------|
|                                  |                  | 1  | AN   | PWM | 16 |              |               |
| Not connected                    | NC               | 1  | AN   | PWM | 16 | NC           | Not connected |
| Reset                            | PC5_TARGET_RESET | 2  | RST  | INT | 15 | NC           | Not connected |
| SPI Chip Select                  | PC3_SPI_SS       | 3  | CS   | TX  | 14 | PA1_UART_TXD | UART transmit |
| SPI Clock                        | PC0_SPI_SCK      | 4  | SCK  | RX  | 13 | PA2_UART_TXD | UART receive  |
| SPI Master Input<br>Slave Output | PC1_SPI_MISO     | 5  | MISO | SCL | 12 | NC           | Not connected |
| SPI Master Output<br>Slave Input | PC2_SPI_MOSI     | 6  | MOSI | SDA | 11 | NC           | Not connected |
| Power supply                     | +3.3V            | 7  | 3.3V | 5V  | 10 | NC           | Not connected |
| Ground                           | GND              | 8  | GND  | GND | 9  | GND          | Ground        |

## Buttons and LEDs

| Designator | Name | Type (LED, BUTTON...) | Description                  |
|------------|------|-----------------------|------------------------------|
| PWR        | PWR  | LED                   | Power Supply Indication LED. |
| LD1        | A    | LED                   | Key A Press Indication LED   |
| LD2        | B    | LED                   | Key B Press Indication LED   |
| LD3        | C    | LED                   | Key C Press Indication LED   |

|     |     |        |                            |
|-----|-----|--------|----------------------------|
| LD4 | D   | LED    | Key D Press Indication LED |
| J1  | PDI | Header | Programming header         |

### Maximum ratings

| Description                 | Min  | Typ | Max | Unit |
|-----------------------------|------|-----|-----|------|
| Power Supply Voltage        | -0.5 | 3.3 | 6   | V    |
| Current into VDD pin        | -    | 100 | 200 | mA   |
| Reset pin Voltage           | -0.5 | 3.3 | 13  | V    |
| I/O pin sink/source Current | -40  | -   | 40  | mA   |

### Programming

Code examples for TouchKey 2 click, written for MikroElektronika hardware and compilers are available on Libstock.

This code snippet initializes the MCU and display, and in an endless loop recognizes which key is being pressed. The key that was touched last is shown on the display.

```

01 void main( void )
02 {
03     system_init();
04     display_init();
05
06     TFT_Set_Font( &HandelGothic_BT21x22_Regular, CL_RED, FO_HORIZONTAL );
07     TFT_Write_Text( "Press Any On-board Key", 50, 120 );
08
09     while( true )
10     {
11         // Key A has been pressed
12         if( key_A_pressed && old_key != KEY_A )
13         {
14             display_key( "A", CL_RED );
15             key_A_pressed = false;
16             old_key = KEY_A;
17         }
18         // Key B has been pressed
19         else if( key_B_pressed && old_key != KEY_B )
20         {

```

```
21     display_key( "B", CL_GREEN );
22     key_B_pressed = false;
23     old_key = KEY_B;
24 }
25 // Key C has been pressed
26 else if( key_C_pressed && old_key != KEY_C )
27 {
28     display_key( "C", CL_BLUE );
29     key_C_pressed = false;
30     old_key = KEY_C;
31 }
32 // Key D has been pressed
33 else if( key_D_pressed && old_key != KEY_D )
34 {
35     display_key( "D", CL_PURPLE );
36     key_D_pressed = false;
37     old_key = KEY_D;
38 }
39 }
40 }
```